移动界程演算理论及应用研究综述 *

林荣德 [1]，江　华 [2†]，黄建新 [1]

(1. 华侨大学 数学科学学院 计算科学福建省高校重点实验室，福建 泉州 362000; 2. 闽南师范大学 粒计算省重点实验室，福建 漳州 363000)

摘　要：移动界程演算通过“界程”这一核心概念来表达有边界的计算场所，并提供界程移动，认证和授权等能力从最基础层次刻画移动计算的本质，成为了移动计算系统形式化理论和应用领域内的重要研究分支。对移动界程演算的理论及应用方面的研究和发展进行了概述，对移动界演算的扩展语义和代数性质的分析方法、移动界演算的空间逻辑和模型检测算法以及移动界程在计算系统建模方面应用现状进行了整理和分析，并对该领域未来进一步研究的方向进行了展望。

关键词：移动界程演算；标号互模拟；上下文观察等价；界程逻辑；模型检测

中图分类号：TP311　　doi: 10.19734/j.issn.1001-3695.2018.10.0722

Overview on theory and applications of ambients calculus

Lin Rongde[1], Jiang Hua[2†], Huang Jianxing[1]

(1. *Fujian Province University Key Laboratory of Computational Science*, *School of Mathematical Science*, *Huaqiao University*, *Quanzhou Fujian* 362000, *China*; 2. *Key Laboratory of Granular*, *Minnan Normal University*, *Zhangzhou Fujian* 363000, *China*)

Abstract: Ambient Calculus, which introduces the central notion of "ambients" to describe the computation area with boundaries, and provides the capabilities of mobility, authentication and authorization to capture the fundamental characteristics of mobile computation systems, becomes an important research area on formal-modeling theory and applications of mobile computation systems. The paper reviews the main theoretical and applied research works and its development of ambient-based calculus, and analyzes the extended semantics and the algebra behavior analyzing methods, the spatial logics and the model checking algorithms, and the applications of formal-modeling in computation systems; and discusses the future research directions on these areas.

Key words: ambient calculus; bisimulation congruences; context barb congruences; ambient logic; model checking

0　引言

移动界程演算（mobile ambients，MA）[1]是 Cardelli 和 Gordon 提出的一种移动计算的进程演算模型。受到 PI 演算的启发，引入了“界程”的这一特殊的进程来刻画有边界的计算场所，如 Web 页面、组件、移动设备等。界程之间按照树型方式组合，每个界程内部包括通信进程、移动进程和子界程。界程受移动进程的控制可以整体移动。移动界程演算通过界程、移动及其与移动相关的授权等概念能够从基础层次刻画分布式及移动计算的本质特征。MA 演算的语法：

$$M::=\epsilon|x|n|in\,n|out\,n|open\,n|M.M'$$
$$P,Q::=0|M.P|n[P]|P\,|\,Q|\langle M\rangle|(x)P|(\nu n)P\,|\,!P \tag{1}$$

在上述定义中，M 表示能力：它可以为空 ϵ、名字变量 x、名字 n、进入界程的能力 $in\,n$、退出界程的能力 $out\,n$ 和打开界程的能力 $open\,n$，或能力的串行组合路径 $M.M'$。

P,Q 表示进程：0 表示钝性进程；$M.P$ 表示执行能力 M 的动作，然后继续进程 P；$n[P]$ 表示名字为 n 内部进程为 P 的界程；$P|Q$ 表示进程 P，Q 并发执行；$\langle M\rangle$ 是异步输出的能力；$(x)P$ 表示输入一个能力 x 然后继续进程 P；$(\nu n)P$ 表示约束名字 n 作为进程 P 内部使用；$!P$ 表示进程 P 的无限重复 $(P|P|\cdots)$。

移动界程演算的语义由结构同余关系 $\equiv$ 和归约关系 $\rightarrow$ 两部分组成（如表 1 和 2 中的规则定义），其中结构同余关系 $\equiv$ 表示语法不同但语义相同的进程，归约关系 $\rightarrow$ 表示进程可能的执行结果。表 2 中归约规则（Red In）表示进入能力 $in\,n$ 可以带领所在界程 m 迁移入另一个同胞的界程 n 内部成为子界程。归约规则（Red Out）表示退出能力 $out\,n$ 可以带领所在界程 m 离开其父界程 n 成为同胞界程。归约规则（Red Open）表示打开能力 $open\,n$ 可以去除界程 n 的边界并暴露其内部进程 Q。归约规则（Red Comm）表示界程本地的通信：输出能力进程 $\langle M\rangle$ 释放能力 M，被输入动作 (x) 捕捉后绑定到进程 P 中的形式参数 x 中。归约规则（Red Amb）（Red Par）和（Red Res）分别表示归约可以在界程内部、并发环境和私有名字空间内发生。归约规则（Red $\equiv$）则表示了满足结构同余关系的进程归约结果是一致的。

从上述 MA 演算的语法和语义定义中可以看出，名字 n 是作为界程名而不像 PI 演算中的通道名。每一个界程都显式地表明了它在系统整个树型结构中的位置，因而每一个进程

收稿日期：2018-10-24；**修回日期：**2018-12-24　**基金项目：**国家自然科学基金资助项目（11871289，61379021，11701258）；福建省自然科学基金资助项目（2016J01304，2015J01269）；福建省高校创新团队发展计划资助项目；泉州市高层次人才团队项目（2017ZT012）

作者简介：林荣德（1967-），男，福建龙岩人，副教授，博士，主要研究方向为形式化方法、计算系统建模；江华（1970-），男（通信作者），湖南醴陵人，副教授，硕导，博士，主要研究方向为形式化方法（sg_jh@126.com）；黄建新（1969-），男，福建仙游人，副教授，主要研究方向为计算系统建模、数据处理.

表达式都可诱导出一个抽象的树型配置（configuration），由此可以很直接地刻画移动计算系统中的局部、父/子资源、边界和系统拓扑等概念。系统的配置又可以通过界程中移动能力的执行、局部通信和资源的动态绑定等方式实现动态改变，特别是由进程通信引起的名字动态绑定实现了移动计算系统隐式重配置，从而能够透明地更新接收方的系统状态。由于上述这些优点，使得 MA 演算成为了描述移动计算系统中普遍存在的进程和资源的位置的分布、进程在不同位置之间的移动，以及相关的安全控制问题的示范性模型。近十几年来，围绕“界程”为核心概念的扩展界程演算模型及应用研究逐渐成为移动计算系统形式化建模理论及应用很活跃的研究领域，主要表现如下几个方面：a）移动界程演算的扩展语义及相关代数理论的研究；b）移动界程空间逻辑及模型检测算法的研究；c）移动界程演算在计算系统的建模应用的研究。

表 1　MA 演算的进程结构同余关系 ≡

Table 1　Structural congruence relations of mobile ambients

（Stru Refl）	$P \equiv P$
（Stru Symm）	$P \equiv Q \Rightarrow Q \equiv P$
（Stru Trans）	$P \equiv QQ \equiv R \Rightarrow P \equiv R$
（Stru Par Assoc）	$P\|(Q\|R) \equiv (P\|Q)\|R$
（Stru Par Comm）	$P\|Q \equiv Q\|P$
（Stru Par Zero）	$P\|0 \equiv P$
（Stru Rep）	$!P\|P \equiv !P$
（Stru Nil）	$\epsilon.P \equiv P$
（Stru Path）	$(M.M').P \equiv M.M'.P$
（Stru Rep）	$P \equiv Q \Rightarrow !P \equiv !Q$
（Stru Res ）	$P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$
（Stru Par）	$P \equiv Q \Rightarrow P\|R \equiv Q\|R$
（Stru Amb）	$P \equiv Q \Rightarrow n[P] \equiv n[Q]$
（Stru Cap）	$P \equiv Q \Rightarrow M.P \equiv M.Q$
（Stru Inp）	$P \equiv Q \Rightarrow (x)P \equiv (x)Q$
（Stru Res Zero）	$(\nu n)0 \equiv 0$
（Stru Res Res）	$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$
（Stru Res Par）	$n \notin fn(Q) \Rightarrow (\nu n)(P\|Q) \equiv ((\nu n)P)\|Q$
（Stru Res Amb）	$n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P]$

表 2　MA 演算的进程归约关系 →

Table 2　Reduction relations of mobile ambients

（Red In）	$m[in\,n.P\|P'] \| n[Q] \to n[m[P\|P']\|Q]$
（Red Out）	$n[m[out\,n.P\|P']\|Q] \to n[P\|P'] \| n[Q]$
（Red Open）	$open\,n.P\|n[Q] \to P\|Q$
（Red Comm）	$(x)P\|\langle M\rangle \to P\{x/M\}$
（Red Amb）	$P \to Q \Rightarrow n[P] \to n[Q]$
（Red Par）	$P \to Q \Rightarrow P\|R \to Q\|R$
（Red Res）	$P \to Q \Rightarrow (\nu n)P \to (\nu n)Q$
（Red ≡）	$P \equiv P', P \to Q, Q \equiv Q' \quad \Rightarrow P' \to Q'$

1　移动界程演算的扩展语义及代数理论研究

移动界程演算所提出的界程及相互之间交互演算反映了移动计算系统的实质，众多研究工作也根据不同角度对移动界程演算进程的交互语义进行了扩展，并研究采用各种形式的互模拟、上下文观察等价等代数理论工具来分析演算模型的各种性质。

1.1　移动界程演算的交互语义的扩展

由于在 MA 演算中界程之间的移动交互原语是单方面参与的，这种设置在语法上将出现强干扰[2]，且无法通过类型系统来识别和消除，导致对 MA 演算的代数理论比较弱，进程的等价性不易判定。许多研究者都从不同方面对 MA 演算进行了扩展，代表性有以安全界程演算[2,3]为代表性的模型、BoxedAmbient 演算[4]和公平界程演算[5]等。

1）安全界程演算

Levi 和 Sangiorgi 首次在文献中指出 MA 演算存在强干扰的问题的同时，提出保留 MA 基本框架，但可以控制强干扰问题的安全界程演算。针对 MA 演算中原语动作的单方参与，为每个原语设置了一个相应的协动作原语。例如针对归约规则（Red In）和（Red Open）分别作如式(2)和(3)所示进行修改：

$$m[in\,n.P|P'] \,|\, n\left[\overline{in}\,n.Q \,|\, Q''\right] \to n[m[P|P']|Q|Q'] \tag{2}$$

$$open\,n.P \big| n\left[\overline{open}\,n.Q \,|\, Q''\right] \to P \big| Q \,|\, Q' \tag{3}$$

通过设置协动作原语，在 MA 出现的强干扰现象就可以通过设置协动作来消除。比如，在 MA 中的进程 $open\,n|n[in\,m.P]\|m[Q]$ 可能出现如式(4)所示的两种互为干扰归约：

$$in\,m.P \,|\, m[Q]$$

或

$$\to open\,n \,|\, m[n[P]|Q] \tag{4}$$

在安全界程演算中通过设置协动作可明确地决定归约方向，从而消除强干扰现象，如式(5)和(6)所示。

$$open\,n|n[in\,m.P]\|m\left[\overline{in}\,m.Q\right] \to open\,n|m[n[P]|Q] \tag{5}$$

或

$$open\,n\big|n\left[\overline{open}\,n.in\,m.P\right]\big\|m[Q] \to in\,m.P \,|\, m[Q] \tag{6}$$

基于安全界程演算为代表的扩展界程演算有认证的交互提高了界程移动交互的安全性，同时也增强了这些界程演算扩展模型的代数性质。然而这些扩展演算中界程之间的通信仍需要通过打开界程才能实现，在某些安全操作方面仍有不足之处。为此，不少学者提出各种跨界程的通信机制的扩展模型，主要以 BoxedAmbient 演算和公平界程演算为代表。

2）BoxedAmbient 演算和公平界程演算

BoxedAmbient 演算是在 MA 演算上的扩展模型，注重于界程演算中的跨边界通信机制。通过为输入输出动作设置通信的方向，明确了界程通信对象。比如界程 n 向外部进程输出规则表示如式(7)所示。

$$\text{(Output}\uparrow\text{)}\ (x)P \,|\, n\left[\langle M\rangle^{\uparrow} Q|Q'\right] \to P\{x/M\} \,|\, n[Q] \tag{7}$$

式(7)通信机制提供了细粒度的界程通信原语，同时由于明确了通信方向，规定了输出资源的接收者。这对于移动计算系统的通信安全的认证和授权行为提供了基础操作。然而 BoxedAmbient 演算由于将 Open 原语从演算中去除将导致移动计算系统中的界程数量可能导致无限膨胀的后果。

公平界程演算（简称 FA 演算）是以界程为核心概念的扩展模型中，安全移动、通信交互操作能力较为完善的演算模型。FA 演算提出了界程交互行为的三项原则：所有交互都是由界程为主体参与的；参与交互的界程相互之间应确认身份；参与交互的界程相互之间得到授权。这种三原则符合了网络环境下移动计算系统安全操作所需的基本要求。主要的归约语义表示如表 3 所示。

表 3 中的（R2）规则的含义是：协动作 $\overline{out}\,n$ 包含在并发界程 a 中，所以主动作 $out\,a$ 的含义为“受外部界程 a 的邀请而需要退出”， $\overline{out}\,n$ 的含义为“希望 n 中的某一子界程的到来”。通信机制（R4）也采用了类似于（R1）中采用相互认证，授权的方式建立跨界程的通信。由于设置了跨界程的通信机制，

FA 演算中的 Open 原语不需要再承担辅助通信的义务，而只是完成将二个界程合并的工作，交互方式也与（R1）一样采用相互认证、授权的方式。

表 3　公平界程演算的进程归约关系 →

Table 3　Reduction relations of fair ambient calculus

R1)　$m[in\ n.P \mid P'] \mid n[\overline{in}\ m.Q \mid Q'] \rightarrow n[m[P \mid P'] \mid Q \mid Q']$
R2)　$n[Q \mid m[out\ a.P \mid P']] \mid a[\overline{out}\ n.R \mid R'] \rightarrow m[P \mid P'] \mid n[Q] \mid a[R \mid R']$
R3)　$m[open\ n.P \mid P'] \mid n[\overline{open}\ m.Q \mid Q'] \rightarrow m[P \mid P' \mid Q \mid Q']$
R4)　$m[n(x).P \mid P'] \mid n[m!y.Q \mid Q''] \rightarrow m[P\{x/y\} \mid P'] \mid n[Q \mid Q']$

与上述其他扩展演算相比，FA 演算具有安全界程演算中保证移动操作的安全机制，也具有像 BoxedAmbient 演算中的跨边界安全通信机制。在某种程度上，FA 演算成为了一种“公平的资源交易”演算：通过通信交互交换市场信息，通过 In、Out 原语去寻找交易对象，最后由 Open 原语进行资源合并而完成交易。

3）其他扩展界程演算

以位置（区域/边界）及其跨边界之间的交互为核心，并根据不同的应用背景对界程演算进行了扩展研究方面，主要包括有上下文敏感的界程演算[6]、多类型边界交互的细胞膜演算[7]和交互行为在时间和概率方面的扩展[8,9]以及界程演算的语义和表达能力研究[10]等。

上下文敏感的界程演算（CCA）是一种由界程和它所处的上下文环境构成的扩展界程演算，将界程及其所处的环境置于同等地位：不但定义了界程之间交互语义，也有显式的界程与环境的交互语义。并在后续的工作[11]研究了一种类型系统来保证上下文敏感的界程演算（CCA）进程执行过程的机密性质;文[12]研究了一种类型系统来保证上下文敏感的界程演算（CCA）进程之间通信方式/进程抽象及上下文环境等的正确性等。同时文[13]提出了具有类似于 CCS 的全局通信的上下文环境敏感的扩展界程演算，并研究了该演算的标号归约语义，观察互模拟等价等性质；类似的工作还有空间敏感的界程演算[14]等。

细胞膜演算是模拟生物学上细胞膜行为的一种扩展界程演算，主要有 Brane 演算[7]和 Membranes 演算[15,16]为代表。细胞膜演算模拟了发生于细胞膜表面的内噬（endocytosis）、胞吐（exocytosis）和分裂（mitosis）等计算行为，后续的研究[17]建立了层次化的膜空间和构造了更加丰富膜移动语义。另外，文献[18]提出了一种基于抽象解释技术来对 Brane 演算（细胞膜演算）的动态行为进行静态分析的方法，并研究了获取细胞膜进程所构造的全局层次结构及相互关系信息的多项式时间算法；文献[19]提出了一种带时间性质的细胞膜演算，给出细胞膜进程在生命时间内的移动性所涉及的各种时空等价性质，并用界程逻辑刻画了位置观察等价性质。

界程演算的交互行为在时间和概率方面的扩展方面，文献[8]提出的概率界程演算在 MA 基础上给出了描述进程的交互动作执行后可能产生多个残留进程以及出现的概率，并提出了观察概率互模拟来刻画进程之间的行为等价的方法，以及在界程逻辑公式中增加概率时态公式；在界程演算交互行为的时间约束的扩展研究中，文献[9]提出了界程是作为资源而存在的：具有持续时间、容量和存取半径等约束以及相应超时处理进程；交互动作也有持续时间约束和相应超时处理进程。另外，文献[20]提出了一种带时间约束的一种扩展界程演算来对移动代理的运行时的时间和空间状态进行描述模型；它可针对进程的带时间约束的动作、动作的持续时间以及空间需求进行形式化描述，并在后续工作[21]提出了一种基于动作的时空上下文模型来对带时间约束和空间约束的动作进行推演分析的方法。

1.2　界程演算的行为等价性理论方法

基于代数语义的进程演算（如 CSP、PI 演算等）中，进程的之间各种等价关系是分析进程行为的重要手段。移动界程演算的进程之间结构同余关系仅仅表达了进程之间静态的和最细粒度的进程等价关系，而在某些情况下却需要更高抽象层次来讨论进程之间的行为等价关系。例如在 MA 演算中的进程 $\nu n[P]$ 表示了一个由私有名字 n 构成一个内部进程为 P 的界程，如果名字 n 也没有在内部进程 P 中出现，则根据表 2 中的归约规则它不能与任何其他进程发生交互：P 不能从界程 n 中出来，其他进程也不可意识到 $\nu n[P]$ 的存在并与之交互，因此从某种行为语义下可视为与钝性进程 0 行为等价：$\nu n[P] \approx 0$。如何合理地定义进程之间的行为语义等价，以及如何建立这种语义下的行为语义等价推演规则？以下针对在扩展界程演算运用标号互模拟、上下文观察等价等代数理论方法来分析和推演移动界程演算进程行为性质的方法进行概述。

1）标号互模拟

标号互模拟是表征进程之间交互行为的等价，它是基于标号转移语义的一种等价关系。一般可如式(8)所示来定义进程 P 和 Q 之间的标号互模拟关系 $\approx$[22]。

$$P \approx Q \text{ 当且仅当对于任何标号 } a \text{:如果 } P \xrightarrow{a} P' \text{ 则存在}$$

$$Q \xrightarrow{\tau}{}^{*} \xrightarrow{a} \xrightarrow{\tau}{}^{*} Q' \text{ 且满足 } P' \approx Q' \tag{8}$$

式(8)中，$P \xrightarrow{a} P'$ 表示进程 P 通过标号 a 的转移成为进程 P'；$Q \xrightarrow{\tau}{}^{*} \xrightarrow{a} \xrightarrow{\tau}{}^{*} Q'$ 表示进程 Q 通过 0 到多次内部转移、标号 a 的转移、再经过 0 到多次内部转移成为进程 Q'。同时，根据不同粒度的行为等价需求可将 $\xrightarrow{a}$ 关系定义为强互模拟[22,23]，或 $Q \xrightarrow{\tau}{}^{*} \xrightarrow{a}$ 关系定义 Early 互模拟[22]等。

标号转移语义是与归约语义相一致的、用于表达单个进程与外部发生交互的可能性。在 PI 演算中所有进程通信交互是在一个平行空间进行，可直接将通信动作定义为标号构造一种提交关系，进而根据该提交关系来构造与归约语义相一致的标号转移系统。而在界程为核心的演算中界程之间构成一个层次空间，界程之间交互动作的执行将导致层次空间的重组，基于提交关系的方法将导致标号转移的结果除了进程，还可能是称为一种称为固化结果(包含了进程中参与归约的部分和剩余部分）[2]。另一种方法是构造一个硬化关系 $>$（hardening）将进程首先硬化成固化结果，然后再由硬化关系和归约语义来构造标号转移系统[24]，并可确保标号转移结果也是进程。

图 1 给出一个在 MA 演算下构造一个与表 2 中（Red In）$m[in\ n.P \mid P'] \mid n[Q] \rightarrow n[Q \mid m[P \mid P']]$ 归约语义相一致的标号转移规则的示例。

图 1 中,硬化规则(Harden Cap)(Harden Par)和(Harden Amb)可以得到硬化关系：$m[in\ n.P \mid P'] \mid n[Q] > \langle m[in\ n.P \mid P'] \rangle n[Q]$，同时由(Harden Amb)可得硬化关系：$n[Q] > \langle n[Q] \rangle 0$，再由(Harden Cap)和(Harden Par)可得 $in\ n.P \mid P' > \langle in\ n.P \rangle P'$；接下来

根据标号转移规则(Trans Cap)可得标号转移：$in\,n.P\,|\,P' \xrightarrow{in\,n} P\,|\,P'$，最后根据(Trans In)得到与表 2 中的归约规则(Red In)相一致的标号转移，如式(9)所示。

$$m[in\,n.P|P'] \mid n[Q] \xrightarrow{\tau} n[Q \mid m[P|P']] \tag{9}$$

$$(\text{Harden Cap})\frac{}{M.P > \langle M.P\rangle 0},\ (\text{Harden Par})\frac{P > (\nu\bar{p})\langle P'\rangle P''}{P \mid Q > (\nu\bar{p})\langle P'\rangle(P'' \mid Q)}$$

$$(\text{Harden Amb})\frac{}{n[P] > \langle n[P]\rangle 0},(\text{Trans Cap})\frac{P > (\nu\bar{p})\langle M.P'\rangle P''}{P \xrightarrow{M} (\nu\bar{p})(P' \mid P'')}$$

$$(\text{Trans In})\frac{P > (\nu\bar{p})\langle m[Q]\rangle R, Q \xrightarrow{in\,n} Q', R > (\nu\bar{r})\langle n[R']\rangle R''}{P \xrightarrow{\tau} > (\nu\bar{p},\bar{r})(n[m[Q']|R'] \mid R'')}$$

图 1　MA 演算下的标号转移规则

Fig. 1　Label translate rule of mobile ambients

构造合理的标号转移系统是各种以界程为核心的扩展演算建立互模拟关系必须要面临的问题。通过定义特定标号可以表达更丰富的行为互模拟关系（如文献[23]中将标号由动作扩展为进程可实现一种更强的互模拟关系），硬化关系将参与交互的进程分割成直接参与部分和作为环境而存在的部分可建立更直观的标号转移系统。在细胞膜演算[7]（一种 MA 演算扩展演算）中利用了硬化关系可实现进程的交互过程进行细化描述这一特点，直接将参与交互的部分定义为膜，实现了细胞膜演算中更复杂的交互行为的表达（详细可参见文献[7]）。

2）上下文观察等价

给定二个进程上下文观察等价是指，将它们置于任何上下文中对进程的观察结果是等价的（即基于观察结果无法区分二个进程）。对进程的观察结果（barbs）是指：给定进程是否能展示（exhibits）指定结果，如指定的界程[24]或界程中潜在交互动作[5,25]等。例如有如式(10)和(11)所示方式来进行定义：

$$P \downarrow n \doteq \exists\vec{p}, P', P''.n \notin \{\vec{p}\}\, P \equiv (\nu\vec{p})(n[P' \mid P'']) \tag{10}$$

$$P \downarrow in\,n \doteq \exists\vec{p}, P', P'', m.n \notin \{\vec{p}\}\, P \equiv (\nu\vec{p})(m[in\,n.P' \mid P'']) \tag{11}$$

分别定义了指定进程可展示一个名字为 n 的界程和可在某个界程中展示一个潜在交互动作 in n。相应地可定义：$P \Downarrow inn$ 和 $P \Downarrow n$ 作为一种弱观察结果：表示进程可经 0 次或多次归约后的观察结果。

在不同的界程扩展演算中观察结果的定义方法可根据该扩展界程演算所确定的归约语义和观察等价的构造需求来确定。如文献[2]将 $P \downarrow n$ 定义 $P \downarrow \overline{inn}$ 或 $P \downarrow \overline{openn}$ 表示进程 P 中有个界程 n 可进行交互（被其他界程进入或打开）。

上下文（context）是由 MA 演算或扩展界程演算所定义的语义规则确定的进程之间的各种组合方式，反映了进程未来将面临的各种交互环境。上下文 $\mathcal{C}\{\}$ 可定义为一个包含 0 个或多个空缺位置（用占位符{}表示）的进程，$\mathcal{C}\{P\}$ 表示 $\mathcal{C}\{\}$ 中每个空缺位置都用进程 P 填满后得到的进程。如果上下文 $\mathcal{C}\{\}$ 的某个空缺位置处于约束名字所辖，原来进程 P 中的自由名字或变量可能成为 $\mathcal{C}\{P\}$ 的约束名或约束变量，比如 MA 演算进程 $P = n[inx.P']$，则 $\mathcal{C}\{P\} = (\nu n)(x)\{P\} = (\nu n)(x)\{n[in\,x.P']\}$ 中的名字 n 和变量 x 是约束的。上下文观察等价 ≃ 一般可按如式(12)的方式来定义：

$$P \simeq Q \doteq \text{对所有观察结果a和任一上下文}\mathcal{C}\{\ \},$$
$$\text{都有}\mathcal{C}\{P\} \Downarrow \text{a当且仅当}\mathcal{C}\{Q\} \Downarrow \text{a} \tag{12}$$

从上述可知，上下文观察等价是从某些分析需求出发，通过界定观察结果和构造相应上下文来分析进程之间行为等价性质；它保证了与结构同余语义之间的一致性，同时也在更抽象的层次上建立进程之间的行为等价关系。

判定或证明进程之间的上下文观察等价关系将需要解决两个主要问题：

a)上下文存在有多个占位，以及进程填充上下文的占位符时可能产生名字捕获（即进程中的自由名字成为约束名的现象）等情况将导致等价性判定过程的复杂性。

通过建立一种简化上下文（Harness），然后再通过推导上下文引理(context lemma)[24]以及结合结构同余和归约规则可进一步得到简化上下文观察等价和一般上下文观察等价是等效的[24]。简化上下文只有一个占位符，同时还规定了约束名在进程填入之前要进行换名，以免发生名字捕获导致等价判定的复杂性。简化上下文可由式（13）所示进行定义：

$$\mathcal{H} ::= \{\} | (\nu n)\mathcal{H} | n[\mathcal{H}] | P | \mathcal{H} | \mathcal{H} | P \tag{13}$$

上下文引理是表明上下文观察等价能够在给定进程的名字空间下对任一名字替换下都保持等价关系的性质，由引理 1 所示。

引理 1　上下文引理（context lemma）。

对于给定进程 P 和 Q，$P \simeq Q$ 当且仅当在域 $dom(\sigma) = fv(P) \cup fv(Q)$ 的任何名字替换 σ，及任何简化上下文 $\mathcal{H}$ 和观察结果 a 下满足：$\mathcal{H}\{P\sigma\} \Downarrow \text{a} \Leftrightarrow \mathcal{H}\{Q\sigma\} \Downarrow \text{a}$

可以看出，引理 1 中的任何名字替换下的上下文观察等价已经包含了一般上下文情况下进程填充占位符时可能产生名字捕获的情景。进一步地，简化上下文观察等价和一般上下文观察等价是等效性可以根据所给定的界程演算或扩展界程演算结构同余和归约语义分别进行归纳证明，可参见文献[26]等。

b)上下文观察结果 $\mathcal{H}\{P\} \Downarrow \text{a}$ 不但要考察 $\mathcal{H}\{P\} \downarrow \text{a}$ 还需要考察 $\mathcal{H}\{P\}$ 的 0 次或多次归约后的观察结果，因此上下文等价的判别需要细粒度地分析上下文及进程的交互过程。类似于标号互模拟方法，文献[24]通过引入上下文的硬化关系，并根据观察结果集来构造与归约语义相一致的标号转移系统，并推导出 $\mathcal{H}\{P\} \Downarrow \text{a}$ 只需要考察引理 2 中的三种情况下 $\mathcal{H}\{P\} \to R$ 归约的观察结果。

引理 2　活动引理(activity lemma)[24]。

如果 $\mathcal{H}\{P\} \to R$，当且仅当如下(a)~(c)之一成立：

(a)上下文 $\mathcal{H}$ 的填充进程 P 内部交互导致的归约：$R \equiv \mathcal{H}\{P'\}$ 其中 $P \to P'$；

(b)上下文 $\mathcal{H}$ 内部交互导致的归约：$R \equiv \mathcal{H}'\{P\}$ 其中 $\mathcal{H} \to \mathcal{H}'$；

(c)上下文 $\mathcal{H}$ 与填充进程 P 之间交互导致的归约：$\mathcal{H}\{P\} \to R \doteq \mathcal{H} \cdot P \to R$。

由此，通过上下文引理和活动引理，实现了利用简化上下文观察等价来等效地表达一般上下文观察等价。

3）其他相关研究

基于标号互模拟和基于上下文观察等价的方法都是在界程演算或扩展界程演算的结构同余和归约语义基础上通过细粒度分析进程之间的交互过程来得到进程潜在行为的等价性，但前者的抽象层次相对比后者弱。因为前者主要从进程本身的结构组成和可能内部归约动作出发分析进程所展示的潜在行为，而后者还需要观察进程置于任何上下文的情况下所展示的潜在行为，其中还包括进程与上下文可能产生交互的情景。然而上下文观察等价的判别意味着需要将进程放入任何必要的上下文中进行观察，因此除了一些如文献[25,26]等所给出几种类型的特定进程，要判定任意进程之间的上下

文观察等价将会相当困难。针对这一问题，文献[5]提出了较弱条件的基于观察结果的行为等价关系：如基于标号转移和观察结果的弱观察互模拟 $\approx_b$[5]，以及在上下文中仅对特定类型的观察结果（不是所有观察结果）所建立的上下文弱观察等价关系 $\approx_b^c$[5]等。

2　移动界程空间逻辑及模型检测算法

2.1　移动界程空间逻辑

在移动计算系统的建模过程中可能需要分析模型的性质，如进程的空间分布结构、移动活性、安全性等。用模态逻辑来描述这些性质是一种很重要的方法，同时也可以借助于模型检测技术实现对这些逻辑性质的自动验证。

相比于传统的并发系统，移动计算系统的空间结构是由众多相对独立的、分散的行为的子系统构成的。传统的模态逻辑如 LTL、CTL*等时序逻辑，或 HML 逻辑无法表达移动计算系统中的多个子系统之间空间结构性质。界程逻辑[27]（简称 AL 逻辑）因能够很好地刻画移动或分布系统的空间结构，成为了示范性的空间逻辑。基本的 AL 逻辑公式的语法如式(14)所示，令 n 表示一个名字或名字变量：

$$A,B::=\mathrm{T}|\neg A|A\vee B|\forall x.A|0|n[A]|A|B|\mathrm{A}@n\,|\,\mathrm{A}\propto\mathrm{B} \\ |\langle\star\rangle A|\prec\star\succ\mathrm{A} \tag{14}$$

由式(14)所示的语法所定义的逻辑公式中，0 表示空间上的钝性；$n[A]$表示空间上位置性质；$A|B$表示二个空间的并列性质；而 $\mathrm{A}@n$ 和 $\mathrm{A}\propto\mathrm{B}$ 分别表示空间位置和空间并列的共轭性质；$\langle\star\rangle A$ 表示时序上某个时刻的性质；$\prec\star\succ\mathrm{A}$ 表示空间上某个位置上的性质；其余公式是传统一阶逻辑公式。

AL 逻辑公式的满足关系 $P\vDash A$ 表示进程 P 满足闭公式 A，相对于 MA 演算的 AL 逻辑公式的满足关系由表 4 定义。对于表示空间上某个位置上的模态词 $\prec\star\succ$，需要定义如式(15)所示的关系 $P\downarrow P'$：

$$P\downarrow P'\doteq\exists n,Q.P\equiv n[P']|Q\ \text{且}\downarrow *\ \text{表示的自反传递闭包} \tag{15}$$

从表 4 可看出，AL 逻辑中空间公式的满足关系严格地依赖于 MA 演算的结构同余关系 $\equiv$，因而借助于 0，$n[A]$和 $A|B$ 等三个基本空间公式就可以对 MA 进程进行直接地、细粒度地刻画。同时通过时间弱模态 $\langle\star\rangle$ 可刻画 MA 进程的归约性质，尤其引入空间位置弱模态 $\prec\star\succ$ 实现了对某一空间位置的观察。即 AL 公式具有刻画界程演算进程在任一时间、任一位置的性质的表达能力。

表 4　AL 逻辑公式相对 MA 演算进程的满足关系

Table 4　Satisfaction relations between AL formula and MA process

$P\vDash\mathrm{T}$　　对于任何 MA 进程都成立
$P\vDash\neg A\doteq not P\vDash A$
$P\vDash A\vee B\doteq P\vDash A or P\vDash B$
$P\vDash\forall x.A\doteq\forall m.P\vDash A\{x/m\}$
$P\vDash 0\doteq P\equiv 0$
$P\vDash n[A]\doteq\exists n,P'.P\equiv n[P'] and P'\vDash A$
$P\vDash A\|B\doteq\exists P',P''.P\equiv P'\|P'' and P'\vDash A and P''\vDash B$
$P\vDash\langle\star\rangle A\doteq\exists P'.P\rightarrow *P' and P'\vDash A$
$P\vDash\prec\star\succ A\doteq\exists P'.P\downarrow *P' and P'\vDash A$
$P\vDash A@n\doteq n[P]\vDash A$
$P\vDash A\propto B\doteq\forall P' P'\vDash A\Rightarrow P\|P'\vDash B$

AL 逻辑的表达能力特别表现在并列共轭模态公式 $A\propto B$，根据满足关系定义，该公式描述了一种测试方式的观察，即对于任何进程 P' 如果满足 A，则与 P 并列组合后都会满足 B。因此 AL 逻辑虽然没有采用类似于 HML 逻辑中用直接观察进程潜在交互行为的模态词 $\langle\alpha\rangle$，但借助于模态词 $\propto$ 通过间接方式来测试观察进程的动作行为性质。

在带不动点算子的界程逻辑方面，林惠民提出了谓词 μ-演算界程逻辑[28]，其不动点逻辑公式可以刻画进程的无限计算行为。该逻辑在 AL 基础上将弱时态词 $\langle\star\rangle$ 修改为表示下一时刻 Next 模态词 $\langle\bullet\rangle$，并通过不动点公式和 Next 模态词 $\langle\bullet\rangle$ 来表达 $\langle\star\rangle$，同时还增加揭示名模态公式 $n®A$ 以及新鲜名量词公式 $\mathbb{N}n.A$。

值得注意的是，谓词 μ-演算界程逻辑可以借助不动点公式刻画如式(16)所示的 AL 逻辑中空间某一位置的弱模态 $\prec\star\succ$：

$$\prec\star\succ A\doteq\mu X.(A\vee\prec\!\bullet\!\succ X) \tag{16}$$

式(16)中谓词变量 X 不在公式 A 中自由出现，空间某一位置强模态 $\prec\!\bullet\!\succ$ 定义为：$\prec\!\bullet\!\succ A=\exists x x[A]|\mathrm{T}$。

另一方面，空间逻辑在模型检测中的可判定性问题是非常重要的，因为它关系到逻辑系统能否被实际使用。AL 逻辑中弱时态模态词 $\langle\star\rangle$ 和并列模态副词 $\propto$ 对 AL 逻辑公式的表达能力影响较大，如果没有弱时态模态词 $\langle\star\rangle$ 的 AL 子逻辑称为 AL 静态子逻辑，其表达能力很有限。文献[29,30]等针对 AL 逻辑的各种子逻辑公式集和不同 MA 演算模型的片断上的可判定性问题进行了研究，主要结论是：

a）在含有有重复操作 $!P$ 的 MA 演算片断、或含并列模态副词 $\propto$ 的 AL 逻辑公式集上的模型检测是不可判定的。

b）在不含有重复操作 $!P$ 的 MA 演算片断，且不含并列模态副词 $\propto$ 的 AL 逻辑公式集上的模型检测是可判定的。

由上述结论可知，除了 MA 演算进程中重复操作 $!P$ 将可能导致进程发生无穷归约行为并导致状态发生无限制的膨胀，可导致模型检测不可判定之外。AL 逻辑中模态词 $\propto$ 是导致模型检测不可判定的关键原因，因为并列模态副词 $\propto$ 将导致无限次测试 MA 中的无限进程集中的进程。然而可判定的 AL 逻辑公式集将因为没有并列模态副词 $\propto$ 其表达能力受到极大的削弱，不能间接地观察进程潜在动作的交互行为。

2.2　移动界程演算模型检测算法

模型检测的状态爆炸问题是由于并发系统下检测算法需要搜索所有并发归约动作交替执行的过程中产生大量状态信息存储的现象，而移动界程演算下的模型检测由于需要同时涉及时态和空间性质的验证，尤其在谓词 μ-演算界程逻辑下检测不动点公式的空间分割性质时，需要考察进程并置交替所产生的所有状态，导致检测过程中的状态空间膨胀程度更加严重。

另一方面，传统基于基于时态逻辑 LTL-x、CTL*-x 或行为模态逻辑 HML、mu-演算等逻辑下所采取的符号化模型检测、偏序归约方法等模型检测算法很难在界程演算模型检测过程直接运用。原因是界程演算下的模型检测中进程的归约过程中，可能引起系统时态性质和空间性质的交替变化，因此界程演算下并发动作执行路径的等价性不但涉及进程本身的归约偏序性质，而且还涉及当前状态下的空间性质是否也满足偏序性[31]。

目前，在移动界程演算不同应用需求下进行模型检测方法的研究方面，文献[27]讨论了未包含重复进程!P 的有穷界程演算和重复进程!P 中未包含界程的有限界程演算的模型检测问题，文献[29]研究了界程演算和界程逻辑下的模型检测算法的复杂性，文献[32]提出了有限控制界程演算并给出了针对有限控制界程演算的模型检测算法。文献[28,33]首次给出了在有限控制界程演算和带不动点算子的界程逻辑下的模型检测算法。

在移动界程演算模型检测的高效算法研究方面，在林惠民基于μ-演算的一阶谓词界程逻辑模型检测算法[33]基础上，江华等人[34,35]提出了算法时间复杂度与界程逻辑公式中不动点算子交替嵌套深度 d 呈指数关系，空间复杂度与 d 呈线性关系的全局模型检测算法，后续工作中[36]研究了基于μ-演算的一阶谓词界程逻辑模型检测计算过程中的中间结果满足的两组偏序关系，提出了将时间复杂性与不动点公式的交错嵌套深度的 1/2 呈指数关系、而空间复杂度与计算节点集的规模呈线性关系的高效局部模型检测算法。

界程演算模型检测算法的其他相关研究还包括，文献[31]研究了界程逻辑下并发算子、界程算子和时态算子交错出现情况下对偏序归约等价性质；文献[26]也研究了界程演算下满足偏序性质的一些单线程进程的充分性问题；以及文献[37]提出了用动作时态逻辑来对移动界程进程进行编码，并进行逻辑行为验证的方法等。

3 移动界程演算在移动计算系统的建模应用

以"界程"为核心概念的移动计算模型能够用界程、移动及其安全等概念从基础层次刻画分布式及移动计算的本质特征。近年来，移动界程演算在云计算服务交互协议及管理系统的建模分析应用，基于模型驱动的软件工程领域的应用，以及作为移动计算程序设计或系统建模语言应用等方面也得到发展。

云计算是一种通过互联网络来提供动态易扩展的虚拟化计算资源服务系统。移动界程演算在云计算领域的建模应用方面的主要研究工作有：文献[38]在移动界程演算基础上提出了虚拟时间界程演算，实现了云计算框架下拥有不同计算能力的嵌套式虚拟机系统进行建模，同时也提出了弱时间互模拟等价下的上下文等价和归约观察同余等云计算模型的分析方法；文献[39]中运用移动界程演算的一个变体—细胞膜演算，针对基于 Dryad 的云计算编程模型进行了形式化建模分析；通过界程的变体—细胞膜中的对象及细胞膜之间的交互来反映执行过程中数据的类型变化，可作为验证程序正确性的辅助工具；文献[40]提出了基于带口令的安全界程演算的一种抽象状态机来对云计算的服务模式、访问控制技术进行了建模分析。文[41]研究了运用界程演算和界程逻辑来实现多域移动网络系统下基于角色的用户授权安全策略建模，并运用模型检测技术对授权安全策略进行验证。此外，文献[42,43]研究了界程演算的扩展演算在网络信息安全传输协议方面的应用；文献[44]提出了一种基于安全界程演算的物联网运行及服务交互协议模型,并且使用界程逻辑对其进行分析验证。

在基于模型驱动的软件工程领域中的应用研究方面，代表性的工作是 Ali 等人[45,46]在传统软件体系结构语言(ADLs)和面向剖面体系结构(aspect-oriented software architecture)中引入界程这一体系结构元素来直观地描述移动对象的位置、交互及跨越边界的行为。Ambient-PRISMA[46]是一个基于.NET 平台的，实现从软件的需求定义、软件体系结构设计以及代码自动生成等功能的模型驱动软件工程项目，在该项目中界程作为体系结构元素表达了组件/连接器位置及它们之间的组合，同时界程也在面向剖面体系结构中表达组件移动功能的行为约束。在该项目的后续研究中，文献[47]运用通道界程演算[48]构造 Ambient-PRISMA 中的组件/连接器的移动行为及约束的形式化验证框架；文献[49]研究了移动应用系统的运行时系统的设备、环境及服务等组件之间的交互所面临的问题，并提出了一种基于界程演算的软件体系结构模型的运行时系统的解决方法；文献[50]则研究了面向剖面软件体系结构下自适应移动资源的位置及移动性的需求刻画方法，同时提出了移动进程根据服务位置和资源位置需求信息实现动态移动配置的算法。上述系列工作建立了一种示范性的，在模型驱动软件工程中实现了移动软件体系结构模型的形式化分析及验证的框架，限于篇幅请参见文献[45~47,49,50]。

以"界程"为核心概念的移动计算程序设计或系统建模语言方面的主要工作有：文献[48]示范性地提出了通道界程系统，它以"界程"演算的扩展模型作为核心程序语言，构造了基于网络环境下移动程序对象在各网络节点进行移动和交互计算的运行时系统。文献[31]研究了界程演算来表达条件选择语义和循环控制的问题，提出的扩展界程演算模型采用了一种标准化操作来处理条件选择结构的进程和带参数递归进程的语义，简化了模型检测时空间逻辑语义的复杂性。文献[51]提出了移动协同资源聚合模型，运用移动界程演算进程对资源进行了自主封装，并利用 MA 的逻辑推理过程，对界程的部分操作及模型中的问题进行了逻辑推理演算等。

4 进一步研究方面

移动界程演算理论的提出是移动计算系统形式化建模理论和应用领域内的重要进展，并经过十几年来的发展来取得丰富的研究成果，但作为面向实际建模应用需要的形式化模型，仍有许多关键问题有待进一步探索和创新。

在移动界程演算的代数理论研究方面，目前基于"互模拟和双引理"框架（即 1.2 节中所述各种互模拟和上下文观察等价的二个关键引理）下的界程演算的行为等价性判别方法的计算复杂度较高[5,24,52~54]，导致它们用于推演移动计算系统的复杂性质时仍然非常困难。因此未来进一步研究可以从两个方面进行创新性探索：

a)在现有理论框架下研究一种高效的过程或步骤来判定所给定进程之间是否存在上下文等价、或更弱些的弱观察互模拟 $\approx_b$、上下文弱观察等价关系 $\approx_b^c$ 或标号互模拟等，以及其他相比于结构同余关系有更高抽象层次的行为等价；同时类型系统是一种可以推演进程之间是否存在共同结构特征的进程静态分析工具，尤其是经过精心构造的类型系统可以识别进程是否存在特定潜在行为[12]；因此可以结合类型推导技术来对界程演算进程进行分类，并针对不同类别采用启发式过程或步骤来减轻判别进程行为等价性的计算复杂度等。

b)基于逻辑的推演系统具有相对的成熟性和表达能力的充分性[54]，因此可以探索一种新的移动界程进程行为的等价性理论框架：将移动界程演算进程映射到基于时序逻辑和空间逻辑的描述及验证框架中，通过在时序逻辑和空间逻辑的框架中的等价性推演来验证移动界程行为性质的等价性。

文献[37,54]所采用的研究方法对上述思路有一定启发性，然而移动界程演算所刻画的行为性质具有时序和空间相互交错的特性，因此研究具有丰富表达能力的，基于时序逻辑和空间逻辑的等价性理论框架是一项具有挑战性工作。

在移动界程的空间逻辑及模型检测研究方面，可判定的 AL 逻辑公式集由于没有并列模态副词 $\propto$ 导致其表达能力受到极大的削弱，不能间接地观察进程潜在动作的交互行为。因此根据应用背景来探索可观察进程行为的界程逻辑方面[55,56]，尤其在界程演算的观察等价理论基础上，满足更高抽象层次上的移动界程行为观察逻辑等是未来重要研究方向。在移动界程模型检测研究方面，面向计算系统建模验证的移动界程模型检测高效核心算法[36]以及实用化的移动界程模型检测工具等应成为主要的研究工作：构造高效且实用化的

界程演算模型检测软件可推动移动界程演算在移动计算系统形式化建模方面深入应用，同时也将推动界程演算模型检测高效算法和软件的进一步研究发展。

移动界程演算在计算系统的建模应用方面，面向实际建模验证所需的可量化实时交互行为性质以及面向系统不确定性的量化分析的概率交互行为等方面研究是进一步推动移动界程演算在实际计算系统的交互协议的建模和分析方面的重要研究工作。同时信息物理系统（cyber-physical systems，CPS）系统中计算实体之间的松耦合性、强自治性和系统时空动态演化性等都表现了一种典型的移动界程演算所刻画的多层次的自治性、交互性等特征[44,57,58]，因此移动界程演算及模型检测技术在 CPS 的交互协议的建模和验证等方面的工作也具有重要的现实意义。

参考文献：

[1] Cardelli L, Gordon A D. Mobile ambients [J]. Theoretical Computer Science, 2000, 240 (1): 177-213.

[2] Levi F, Sangiorgi D. Controlling interference in ambients [C]// Proc of Conference Record of the Annual ACM Symposium on Principles of Programming Languages. New York: ACM Press, 2000: 352-364.

[3] Bugliesi M, Castagna G. Secure safe ambients [J]. Symposium on Principles of Programming Languages, 2001, 36 (3): 222-235.

[4] Bugliesi M, Castagna G, Crafa S, *et al.* Boxed ambients [C]// Proc of International Symposium on Theoretical Aspects of Computer Software. Berlin: Springer-Verlag, 2001: 38-63.

[5] Fu Yuxi. Fair ambients [J]. Acta Informatica, 2007, 43 (8): 535-594.

[6] Siewe F, Zedan H, Cau A, *et al*. The calculus of context-aware ambients [J]. Journal of Computer and System Sciences, 2011, 77 (4): 597-620.

[7] Cardelli L. Brane calculi [C]// Proc of International Conference on Computational Methods in Systems Biology. Berlin: Springer-Verlag, 2004: 257-278.

[8] Kwiatkowska M Z, Norman G, Parker D, *et al.* Probabilistic mobile ambients [J]. Theoretical Computer Science, 2009, 410 (12): 1272-1303.

[9] Ciobanu G. Interaction in time and space [J]. Electronic Notes in Theoretical Computer Science, 2008, 203 (3): 5-18.

[10] 龙环. 灰箱演算的操作语义及表达能力研究 [D]. 上海: 上海交通大学, 2009. (Long Huan. On the semantics and expressiveness of ambient calculi [D]. Shanghai: Shanghai Jiao Tong University, 2009.)

[11] Siewe F. A privacy type system for context-aware mobile ambients [J]. Procedia Computer Science, 2015, 52 (1): 98-105.

[12] Pasqualin D P, Vizzotto J K, Piveta E K, *et al*. Typed context awareness ambient calculus for pervasive applications [J]. Formal Aspects of Computing, 2015, 27 (5): 885-916.

[13] Gul N. A Calculus of mobility and communication for ubiquitous computing [C]// Proc of International Workshop on Automated Specification and Verification of Web Systems. [S. l.] : Open Publishing Association, 2015: 6-22.

[14] Barbanera F, Bugliesi M, Dezaniciancaglini M, *et al.* Space-aware ambients and processes [J]. Theoretical Computer Science, 2007, 373 (1): 41-69.

[15] Ciobanu G, Aman B. on the relationship between membranes and ambients [J]. BioSystems, 2008, 91 (3): 515-530.

[16] Paun G. membrane computing and brane calculi (some personal notes) [J]. Electronic Notes in Theoretical Computer Science, 2007, 171 (2): 3-10.

[17] Aman B, Ciobanu G. describing the immune system using enhanced mobile membranes [J]. Electronic Notes in Theoretical Computer Science, 2008, 194 (3): 5-18.

[18] Bodei C, Brodo L, Gori R, *et al.* A static analysis for brane calculi providing global occurrence counting information [J]. Theoretical Computer Science, 2017, 696 (10): 11-51.

[19] Aman B, Ciobanu G. Behavioural observations of cell movements with timing aspects [J]. Nano Communication Networks, 2015, 6 (3): 96-102.

[20] Boukharrou R, Ilie J-M, Saidouni D E, *et al*. Spatio-temporal planning for mobile ambient agents [J]. Procedia Computer Science, 2015, 56 (1): 96-103.

[21] Boukharrou R, Ilie J M, Saidouni D E, *et al*. Contextual time reasoning for mobile ambient agents [J]. International Journal of Wireless and Mobile Computing, 2016, 10 (3): 250-260.

[22] Merro M, Hennessy M. Bisimulation congruences in safe ambients [J]. Symposium on Principles of Programming Languages, 2002, 37 (1): 71-80.

[23] Jiang Hua, Tan Xinxing. Bisimulations in the boxed safe ambients with password [C]// Proc of International Conference on Information Technology: New Generations. [S. l.] : IEEE Computer Society, 2009: 443-448.

[24] Gordon A D, Cardelli L. Equational properties of mobile ambients [J]. Mathematical Structures in Computer Science, 2003, 13 (3): 371-408.

[25] Vigliotti M G, Phillips I. Barbs and congruences for safe mobile ambients [J]. Electronic Notes in Theoretical Computer Science, 2002, 66 (3): 37-51.

[26] 管旭东，杨怡玲，尤晋元. 移动灰箱演算中强干扰问题的进一步控制 [J]. 软件学报, 2002, 13 (5): 1018-1023. (Guan Xudong, Yang Yiling, You Jinyuan. Further control on the grave interference in mobile ambients [J]. Journal of Software, 2002, 13 (5): 1018-1023.)

[27] Cardelli L, Gordon A D. Anytime, anywhere: modal logics for mobile ambients [C]// Proc of Conference Record of the Annual ACM Symposium on Principles of Programming Languages. New York: ACM Press, 2000: 365-377.

[28] Lin Huimin. Predicate μ-calculus for mobile ambients [J]. Journal of Computer Science and Technology, 2005, 20 (1): 95-104.

[29] Charatonik W, Dalzilio S, Gordon A D, *et al.* The complexity of model checking mobile ambients [J]. Foundations of Software Science and Computation Structure, 2001: 152-167.

[30] 颜锋，陈韬略，韩婷婷，等. 空间逻辑的一个定义框架及其可判定性 [J]. 计算机科学, 2006, 33 (6): 7-10. (Yan Feng, Chen Taolue, Han Tingting, *et al.* A definition framework of spatial logic and decidability [J]. Computer Science, 2006, 33 (6): 7-10.)

[31] 林荣德. 移动界程演算及模型检测应用的关键问题研究 [D]. 广州: 华南理工大学, 2010. (Lin Rongde. Research on key issues of mobile ambients and model checking applications [D]. Guangzhou: South China University of Technology, 2010.)

[32] Charatonik W, Gordon A D, Talbot J M. Finite-control mobile ambients. [C]// Proc of European Symposium on Programming Languages and Systems. Berlin: Springer-Verlag, 2002: 295-313.

[33] 林惠民. 移动进程的空间逻辑 [J]. 中国科学 E 辑: 技术科学, 2004, 47 (3): 394-408. (Lin Huimin. A predicate spatial logic for mobile processes [J]. Science in China Serise F: Information Science, 2004, 47 (3): 394-408.)

[34] 江华. 命题 μ -演算全局模型检测的高效算法设计 [J]. 计算机研究

与发展, 2010, 47 (8): 1424-1433. (Jiang Hua. Efficient global model-checking for propositional μ-calculus [J]. Journal of Computer Research and Development, 2010, 47 (8): 1424-1433.)

[35] 江华，李祥. 移动界程模型检测 [J]. 计算机研究与发展, 2009, 46 (10): 1750-1757. (Jiang Hua, Li Xiang. Model checking for mobile ambients [J]. Journal of Computer Research and Development, 2009, 46 (10): 1750-1757.)

[36] 江华. 基于偏序规律的μ-演算一阶谓词界程逻辑模型检测 [J]. 计算机学报, 2016, 39 (12): 2547-2561. (Jiang Hua. Model checking for first-order predicate ambient logic based on μ-calculus with partial orders [J]. Chinese Journal of Computers, 2016, 39 (12): 2547-2561.)

[37] Aman B, Ciobanu G. Expressing mobile ambients in temporal logic of actions [C]// Proc of Romanian Academy Series a-Mathematics Physics Technical Sciences Information Science. Bucharest: Editura ACAD Romane, 2014, 15 (1): 95-104.

[38] Johnsen E B, Steffen M, Stumpf J B. Virtually timed ambients: a calculus of nested virtualization [J]. Journal of Logical and Algebraic Methods in Programming, 2018, 94 (1): 109-127.

[39] 刘磊，刘丰，任俊绮，等. 基于细胞膜演算的 Dryad 形式化描述 [J]. 哈尔滨工程大学学报, 2016, 37 (11): 1539-1545. (Liu Lei, Liu Feng, Ren Junqi, *et al.* A formal description method of dryad using membrane calculus [J]. Journal of Harbin Engineering University, 2016, 37 (11): 1539-1545.)

[40] 卢晨. 基于扩展 ambient ASM 的云计算形式化模型 [D]. 南宁：广西民族大学, 2016. (Lu Chen. A cloud computing model based on extended ambient ASM [D]. Nanning: Guangxi University for Nationalities, 2016.)

[41] Unal D, Caglayan M U. XFPM-RBAC: XML-based specification language for security policies in multidomain mobile networks [J]. Security and Communication Networks, 2013, 6 (12): 1420-1444.

[42] Jiang Hua, Li Xiang. Boxed Safe Ambients with password and simulation of e-mail system [C]// Proc of IEEE International Symposium on Information Technologies and Applications in Education. [S. l.] : IEEE Computer Society, 2007: 337-342.

[43] Jiang Hua, Tan Xinxing, Li Xiang. Boxed safe ambients with password and application on the Internet [C]// Proc of IEEE International Conference on Networking, Sensing and Control. [S. l.] : IEEE Computer Society, 2008: 472-477.

[44] 丛新宇，虞慧群. 基于安全灰箱演算的物联网移动性建模验证 [J]. 华东理工大学学报：自然科学版 , 2015, 41 (3): 391-395. (Cong Xinyu, Yu Huiqun. Modeling and verification of mobility in cyber physical systems based on mobile safe ambients [J]. Journal of East China University of Science and Technology :Natural Science Edition , 2015, 41 (3): 391-395.)

[45] Ali N, Millan C, Ramos I, *et al.* Developing mobile ambients using an aspect-oriented software architectural model [C]// Proc of International Conference on Move to Meaningful Internet Systems. Berlin: Springer-Verlag, 2006: 1633-1649.

[46] Ali N, Ramos I, Solis C, *et al.* Ambient-PRISMA: ambients in mobile aspect-oriented software architecture [J]. Journal of Systems and Software, 2010, 83 (6): 937-958.

[47] Ali N, Tuosto E. Architectural models of ambient-PRISMA in channel ambient calculus [C]// Proc of IEEE Software Engineering Workshop. [S. l.] : IEEE Computer Society, 2012: 1-10.

[48] Phillips A, Yoshida N, Eisenbach S, *et al.* A distributed abstract machine for boxed ambient calculi [J]. Lecture Notes in Computer Science, 2004, 2986: 155-170.

[49] Ali N, Solis C. Mobile architectures at runtime: research challenges [C]// Proc of International Conference on Mobile Software Engineering and Systems. New York: ACM Press, 2014: 41-44.

[50] Ali N, Solis C. Self-adaptation to mobile resources in service oriented architecture [C]// Proc of IEEE International Conference on Mobile Services. [S. l.] : IEEE Computer Society, 2015: 407-414.

[51] 郝艳梅. 移动协同资源聚合模型研究 [D]. 石家庄：河北经贸大学, 2014. (Hao Yanmei. Research on aggregation model of mobile collaboration [D]. Shijiazhuang: Hebei University of Economics & Business, 2014.)

[52] Busi N, Zavattaro G. Deciding reachability in mobile ambients [C]// Proc of the 14th European Symposium on Programming. London: Springer-Verlag, 2005: 248-262.

[53] Maffeis S, Phillips I. On the computational strength of pure ambient calculi [J]. Theoretical Computer Science, 2005, 330 (3): 501-551.

[54] 魏峻，冯玉琳. 移动计算形式理论分析与研究 [J]. 计算机研究与发展, 2000, 37 (2): 129-139. (Wei Jun, Feng Yulin. Analysis of formal models and methods on mobile computing [J]. Journal of Computer Research and Development, 2000, 37 (2): 129-139.)

[55] 林荣德，奚建清，郭玉彬. 移动进程的蛰伏性及空间逻辑 [J]. 计算机科学, 2009, 36 (3): 173-178. (Lin Rongde, Xi Jianqing, Guo Yubin. Dormancy and spatial logic of mobile ambients [J]. Computer Science, 2009, 36 (3): 173-178.)

[56] 陈江，林荣德. 具有行为观察的移动界程演算空间逻辑 [J]. 南京师范大学学报 :工程技术版 , 2011, 11 (4): 70-76. (Chen Jiang, Lin Rongde. Spatial logic with behavioral observations for ambient calculus [J]. Journal of Nanjing Normal University :Engineering and Technology Edition , 2011, 11 (4): 70-76.)

[57] 彭超宇. 基于模型检验的信息物理融合系统安全性研究 [D]. 南京：南京邮电大学, 2015. (Peng Chaoyu. Research on security of model checking-based cyber-physical systems [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2015.)

[58] Lanotte R, Merro M. A semantic theory of the Internet of things [J]. Information and Computation, 2018, 259: 72-101.